Unit
M

Implementing
Advanced DHTML Features

Objectives

► **Understand advanced features**
► **Filter content**
► **Scale content**
► **Animate element position**
► **Create 3D animation**
► **Transition elements**
► **Create a slideshow**
► **Transition between pages**

Using tools such as CSS, dynamic style, dynamic content, and CSS positioning, you can create interactive Web pages with print-quality formatting and layout. By combining these tools and writing scripts to work with them, you can continually create new features. The companies making browsers also continue to simplify the process by adding browser-native features that require little code and no scripting. Lydia Burgos's supervisor has asked her to create a Web-based presentation about Nomad Ltd. In addition to incorporating CSS formatting and positioning to make the page attractive and easy to read, she plans to add advanced effects to keep her users' interest.

Understanding Advanced Features

In addition to the many effects that the basic DHTML tools offer, you can create complex-looking visual features through a combination of proprietary effects and simple scripts. Although overuse of any dynamic feature can overload users visually and slow down their computers, the limited and precise use of these advanced features can help users focus on the most important aspects of each page. Many proprietary advanced features work only on Internet Explorer 4. However, these innovations will likely be supported by both fifth-generation browsers. Learning these features now is a good way to stay current with trends in Web page design. ✐ As Lydia organizes her presentation, she notes different advanced features that she can use for different effects.

Details

Modifying an element's appearance

In addition to basic color and sizing formatting for text and graphics alike, Internet Explorer offers predefined element formats that affect appearance in complex ways. These formats, known as **filters**, allow you to create many effects, such as a shadow or glow, as shown in Figure M-1. Another way to affect element appearance is by combining a script with CSS position or size information to create the effect of movement, or **animation**. By slowly changing an element's placement or size with a script, you can create the effect of movement without requiring special software or extensive system resources. To help ensure your page layouts remain attractive at different screen resolutions and browser window sizes, you also can include simple scripts to resize elements, depending on each user's browser size.

Open and close effects

You can effectively draw attention to a particular page element by scripting its appearance when the page opens. Internet Explorer offers predefined effects called **transitions**, which cause elements to appear gradually and in specific patterns when the page opens or exits. You can apply these effects to selected elements or to the entire page, as shown in Figure M-2. By using an animation script with offscreen starting coordinates and a timer, you can create **presentation effects** in which elements appear on the screen gradually and in a specific order.
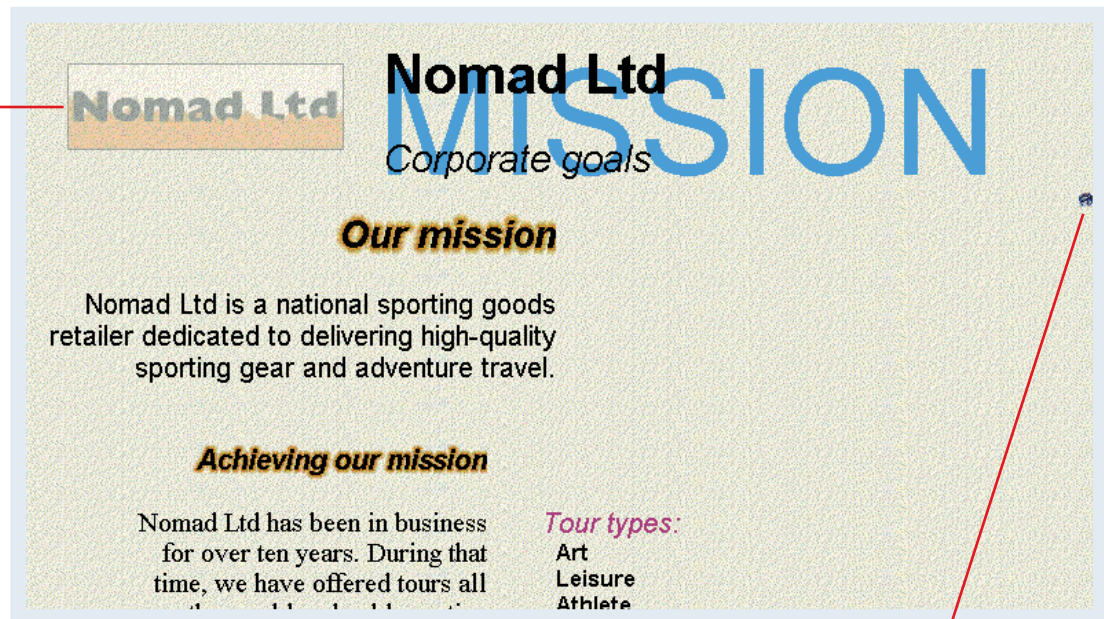
Glow filter adds
colored halo to text

Graphic appearing
gradually using
"blend" transition

Earth graphic as it
appears at start of
load

Filtering Content

DHTML includes several tools that let you change the basic appearance of text and graphics by varying the size, color, and other characteristics of selected elements. In addition, Internet Explorer supports an extended set of properties, known as **filters**, that allow you to modify element appearance in complex ways. Table M-1 lists and describes the filters available in Internet Explorer 4. Lydia wants to call attention to headings in her presentation page without adding more colored text to the page, so she decides to try filtering the headings instead.

Steps

1. Start your text-editor program, open **HTML M-1.htm**, then save it as a text document with the filename **Presentation filter.htm**

2. Select the text **[replace with misshead filter code]** in the embedded style sheet, then press **[Delete]**

3. Type **{height: 14pt; filter: glow(color=#B8860B)}**
   To apply a filter using a DIV or SPAN tag, the text must be absolutely positioned or have a defined height or width. In order to meet this requirement without affecting her layout, Lydia sets the height to 14pt, which is the same height as the heading text.

4. Select the text **[replace with second filter code]** in the #achieve style definition, then press **[Delete]**

5. Type **height: 12pt; filter: glow(color=#B8860B, strength=3)**
   Because the font size in the second heading is smaller than in the first, Lydia decides to lower the intensity of the glow by assigning a strength value. This keeps the glow effect proportional to the text dimensions. Figure M-3 shows the completed code for the glow filters in the embedded style sheet.

6. Scroll down until the <H2> heading "Our mission" appears in the document window
   Lydia wants to apply the filter to the text within the H2 tags. However, filters are incompatible with all heading tags, so she has to embed the H2 tags within DIV tags and then call the filter from the opening DIV tag instead.

7. Type **<DIV ID="misshead">** before <H2> Our mission </H2>, then position the insertion point after the closing H2 tag and type **</DIV>**

8. Check your document for errors, make changes as necessary, then save **Presentation filter.htm** as a text document

9. Open **Presentation filter.htm** in your Web browser
   See Figure M-4, which shows the presentation Web page containing the filtered text. If you are using Internet Explorer 4, the filter adds a halo of color around each letter.

CLUES TO USE

Combining filters

In addition to the unique effect that each filter can create on a Web page, you can increase the possibilities by combining filters, a process known as **chaining**. You can add as many filters as you desire to an element by simply listing them in the element's tag or style sheet description. You must separate the code for each filter by a space. For example, you can chain a drop shadow and a glow to a graphic by calling both of these filters in the picture's style sheet description as follows: **#image: {position: absolute; top: 150px; filter: dropshadow(color=#483D8B, OffX=3, OffY=3) glow(color=#9933CC, strength=5)}**. This code would add both drop shadow and glow effects to the graphic.

FIGURE M-3: Web document containing filter code

Style code to add
filter effect to text

```
#heading {position: relative; left: 250px}
#mission {width: 375px; font-family: arial}
#misshead {height: 14pt; filter: glow(color=#B8860B)}
#main {position: absolute; left: 60px; width: 275px}
#earth {float: right; position: relative; top: -20px}
.norm {font-weight: normal}
#head1 {font-style: italic; color: #8E236B}
#head2 {font-style: italic; color: #6B8E23}
#head3 {font-style: italic; color: #7093DB}
#head4 {font-style: italic; color: #9400D3}
#achieve {height: 12pt; filter: glow(color=#B8860B, strength=3)}
.font12 {font-size: 12pt}
.bot {color: #3232CD}
.space {line-height: 6pt; position: relative}
.item {position: relative; left: 10px; font-size: 10pt; font-weight: bold}
UL {position: relative; left: -20px; top: -20px; font-size: 10pt; font-
weight: bold}
.sidebar {float: right; position: absolute; left: 375px; font-family: arial;
width: 200px}
</STYLE>

</HEAD>

<BODY BACKGROUND="Egg shell.jpg">

<DIV ID="logo">
```

FIGURE M-4: Web page displaying filtered text

Default filter
with color

Reduced filter
strength matches
smaller text

TABLE M-1: Internet Explorer 4 filter effects

filter effect	description	filter effect	description
Alpha	Sets a transparency level	Grayscale	Drops color information from the image
Blur	Creates the impression of moving at high speed	Invert	Reverses the hue, saturation, and brightness values
Chroma	Makes a specific color transparent	Light	Projects light sources onto an object
Drop Shadow	Creates an offset solid silhouette	Mask	Creates a transparent mask from an object
FlipH	Creates a horizontal mirror image	Shadow	Creates a solid silhouette of the object
FlipV	Creates a vertical mirror image	Wave	Creates a sine wave distortion along the x- and y-axes
Glow	Adds radiance around the outside edges of the object	XRay	Shows just the edges of the object

Scaling Content

One drawback of using basic DHTML positioning is a layout's reliance on a particular window size. For example, an image may fit well in a layout at a certain indentation in a maximized browser window on an SVGA screen. However, on a lower-resolution monitor or on a non-maximized browser window, the element may appear much closer to the right edge of the screen, thus changing the original layout design. Using basic scripts to complement CSS-P, however, you can automatically adjust the position of your Web page elements based on the browser window size. Because Navigator does not recognize changes in style properties (including element dimensions) after the page has loaded, this feature works only in Internet Explorer. ▰ Lydia has laid out her page in a maximized browser window set at a resolution of 800 × 600. However, she wants her layout to remain as consistent as possible in smaller windows and at lower resolutions.

Steps 1234

1. Open **HTML M-2.htm** in your text editor, then save it as a text document with the filename **Presentation scale.htm**

2. Scroll down to the script tags in the page's head section, select the text **[replace with scale script]**, then press **[Delete]**
   Lydia has already inserted a browser-detection script, along with a line of code to suppress errors in Navigator.

3. Type the following script, pressing **[Enter]** at the end of each line:

```
function change() {
     if (document.body.clientWidth < 640) {
               bgword.style.fontSize="48pt"
               earth.style.width="25%"
     }
     else {

               bgword.style.fontSize="64pt"
               earth.style.width="35%"
     }
}


window.onresize=change
window.onload=change
```
   Instead of trying to adjust every screen element to fit on a smaller screen, Lydia focuses on the elements along the right edge of the screen: the heading background text and the earth graphic. Her script adjusts both elements to a size that fits into a maximized browser window set at a 640 × 480 resolution. Figure M-5 shows the Web page document code containing the script.

4. Check your document for errors, make changes as necessary, then save **Presentation scale.htm** as a text document

5. Open **Presentation scale.htm** in your Web browser, then make sure your browser window is maximized

6. If you are using Internet Explorer and your display mode is 800 × 600 or greater, click the **Restore Window button** 🗗 at the top right of the browser window to decrease the size of the document window a fixed amount
   If your display is in 640 × 480 mode, note that the large background text and the earth graphic fit on the screen without requiring you to scroll right. Figure M-6 shows the presentation Web page in a reduced window. The scale script you inserted reduced the background text size and the graphic size to fit better in a limited display area.

FIGURE M-5: Web page document code containing scaling script

```
<SCRIPT LANGUAGE="javascript">
<!--
Nav4 = (document.layers) ? 1:0;
IE4 = (document.all) ? 1:0;

if(!IE4) {window.onerror=null}

function change() {
        if (document.body.clientWidth < 640) {
                bgword.style.fontSize="48pt"
                earth.style.width="25%"
        }
        else {
                bgword.style.fontSize="64pt"
                earth.style.width="35%"
        }
}

window.onresize=change
window.onload=change
//-->
</SCRIPT>

</HEAD>

<BODY BACKGROUND="Egg shell.jpg">
```

Script to resize elements based on window size

FIGURE M-6: Scaled objects in reduced browser window

Browser window size reduced from maximized view

Reduced text and graphic sizes fit in window

CLUES TO USE

Scaling by percent

Specifying element dimensions in percentages, rather than pixels or points, has many applications in DHTML design. Usually, you can simply specify the height, width, or font size in percent. Because percentage measurements reflect a percentage of the parent element dimension, a percentage-sized element automatically resizes when the window size changes. To make sure the element remains proportionally scaled when specifying element dimensions, be sure to specify only height or width, but not both. Sometimes, screen elements need to change position depending on the screen size or when an element such as a graphic would look distorted if it became too big or too small. In these cases, you need a scaling script to resize your pages.

Animating Element Position

By creating simple scripts to interact with position and layer information, you can add impressive features to your Web pages without requiring extensive system resources on a user's computer. To create basic animation, for example, you can script an element's position coordinates to increase or decrease slowly when the user first opens the page, until the element reaches its final, absolute coordinates. ⬤▬▬ Lydia decides to animate the Nomad Ltd logo to move into place when a user first opens the page.

Steps 1 2 3 4

1. Open the file **HTML M-3.htm** in your text editor, then save it as a text document with the filename **Presentation position animate.htm**

2. Scroll down to the **function slide()** in the head section and examine the script
   The function slide() positions the logo graphic out of screen range on the right side of the page and then incrementally reduces its left coordinate until it reaches the final position of 30. Lydia also has changed the left coordinate for the #logo style to -1000, a value that triggers the slide() function.

3. Scroll down to the opening BODY tag, select the text **[replace with event handler]**, then press **[Delete]**

4. Type **onLoad="slide()"**
   The onLoad event handler triggers the "slide" script every time the browser loads the BODY section. Figure M-7 shows the code for the event handler to call the slide() function.

5. Check your document for errors, make any necessary changes, then save **Presentation position animate.htm** as a text file

6. Open **Presentation position animate.htm** in your Web browser
   As Figure M-8 shows, the graphic slides into position from the right edge of the window after you open the page. Because Navigator can't change a page's style information after loading, it does not display the logo in its final location.

```
window.onresize=change
window.onload=change

function slide() {
        var pic = document.all.logo;
        if (-1000 == pic.style.pixelLeft) {
                pic.style.pixelLeft = document.body.offsetWidth +
document.body.scrollLeft;
        }
        if (50 <= pic.style.pixelLeft) {
                pic.style.pixelLeft -= 20;
                setTimeout("slide();", 50);
        }
        else {pic.style.pixelLeft =30;}
}
//-->
</SCRIPT>

</HEAD>

<BODY BACKGROUND="Egg shell.jpg" onLoad="slide()">

<DIV ID="logo">
<IMG SRC="nomad.jpg">
</DIV>
```

Slide function in page head

Event handler triggers slide script when page opens

FIGURE M-8: Nomad Ltd logo sliding into position

Nomad Ltd logo sliding right to left into final position

HTML

Creating 3D Animation

You can easily create simple animation on your Web pages with a script that slowly adjusts an element's top or left attribute over a period of time. By incorporating changes in element size, using the width and height properties, you also can create the illusion of 3D movement. Although animation in standard multimedia formats can require special software or browser extensions and significant computer memory, DHTML animation creates the effect of movement using just one image and a small script running on the user's browser. A lot of animation could distract users from the rest of your Web page, but a short animation, or animation of a small element, can make a page interesting and distinctive. ✎ Lydia decides that instead of having the Nomad Ltd logo graphic move into position sideways, she would like the earth graphic to appear to approach the user. She creates this effect with 3D animation.

Steps 1 2 3 4

1. Open the file **HTML M-4.htm** in your text editor, save it as a text document with the filename **Presentation 3D animate.htm**

2. Scroll down and select the text **[replace with 3D animation script]** in the page's head section, press **[Delete]**, then type the following script, pressing **[Enter]** at the end of each line:

```
function grow() {
    if (earthpic.width<250) {
            x=window.setTimeout('grow()', 100)
            earthpic.width=earthpic.width + 10
    }
}

window.onload = grow;
```

Figure M-9 shows the Web page code containing the script. The script uses the graphic's HTML width property, rather than the CSS width, because HTML width is easier to work with in this situation. Lydia has deleted the logo animation script.

3. Scroll down to the IMG tag for the earth graphic, and replace the text **[replace with width property]** with **WIDTH=0**

The script you entered increases the width value by 10 pixels at a time and pauses for a fraction of a second between each increase, which creates the illusion of animation.

4. Check your document for errors, make any necessary changes, then save **Presentation 3D animate.htm** as a text document

5. Open **Presentation 3D animate.htm** in your Web browser

Figure M-10 shows the page as it is loading. As the page loads, the earth graphic appears and slowly grows as it seems to move toward you.

FIGURE M-9: Web document containing grow script

3D animation script
for earth graphic

```
window.onload=change

function grow() {
        if (earthpic.width<250) {
                x=window.setTimeout('grow()', 100)
                earthpic.width=earthpic.width + 10
        }
}

window.onload = grow;
//-->
</SCRIPT>

</HEAD>

<BODY BACKGROUND="Egg shell.jpg">

<DIV ID="logo">
<IMG SRC="nomad.jpg">
</DIV>

<DIV ID="bgword">
MISSION
</DIV>

<DIV ID="heading">
```
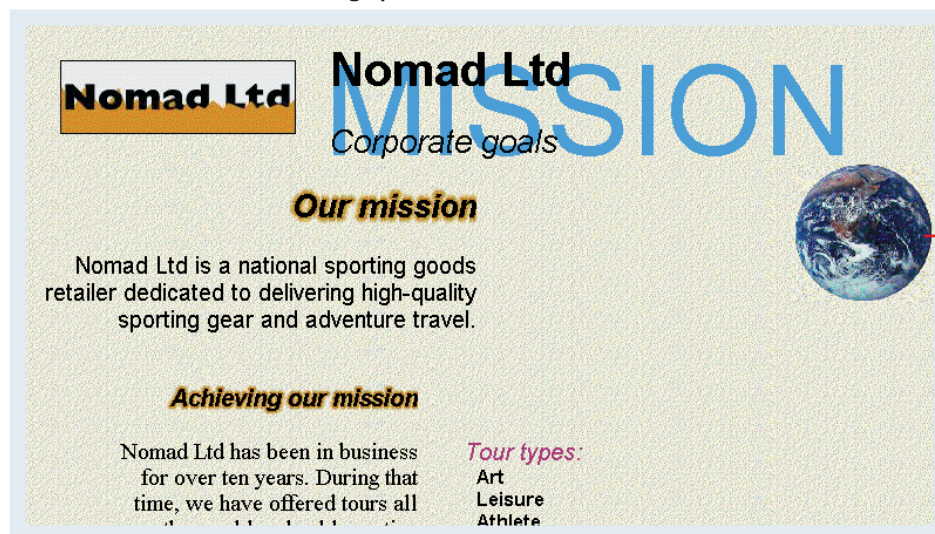
FIGURE M-10: 3D animation of earth graphic

Gradual size
increase on load
creates illusion of
earth approaching

CLUES TO USE

Animated GIFs

Another popular way to create animation in Web pages is to create **animated GIFs**. A GIF is a graphic file in a specific format. Although most GIFs are static, showing just one image, the GIF format also supports animation. To create an animated GIF, you use special software to combine two or more static graphics and to specify the delay between the display of each frame. You can create a movie-like movement effect with animated GIFs, but one of their most widespread uses on the Web today is in banner advertisements on Web pages. These static GIFs often alternate between two different frames of information, such as an advertising motto and the company logo. Although DHTML animation is easier and cheaper to create, animated GIFs are not limited to fourth-generation browsers and are thus accessible by a wider Web audience.

Transitioning Elements

Beyond simply hiding and showing an element, or applying filters to an element's display, you can affect the way an element becomes visible or hidden by using filter effects known as **transitions**. For example, one popular transition effect is to make an element appear or disappear gradually in a checkerboard pattern. Because Navigator does not recognize transitions, this feature works only with Internet Explorer. Internet Explorer 4 comes with two transition filters: **blend**, which creates a simple fade-in or fade-out effect, and **reveal**, which allows the more complex filtering effects. These effects, which can be applied to text as well as graphics, can keep a user's interest and distinguish your pages from others on the Web. ✐ Lydia decides to use the blend transition on the Nomad Ltd logo when the page opens.

Steps 1 2 3 4

1. Open the file **HTML M-5.htm** in your text editor, then save it as a text document with the filename **Presentation transition element.htm**

2. Select and replace the text **[replace with style information]** which is in the #logo style specification in the page's embedded style sheet with **visibility: hidden; filter: blendTrans(duration=7)**
The blend transition can switch from hidden to visible or vice versa. Lydia specifies that she wants the graphic to start out hidden and then become visible using the blendTrans filter. The duration variable details the length of time in seconds of the transition from beginning to end.

3. Scroll down to the end of the page's head section, then replace the text **[replace with transition function]** with the following script, pressing **[Enter]** at the end of each line:

```
function doTrans() {
    logo.filters.blendTrans.Apply();
    logo.style.visibility="visible";
    logo.filters.blendTrans.Play();
}
```

Unlike standard filters, transition filters require scripts to define what happens when they run. The first line of the doTrans() function calls the transition's Apply method, which creates the final state defined in the next line, which is "visible." Finally, the Play method starts the transition filter itself to create the smooth change from hidden to visible. Figure M-11 shows the Web document code containing the function.

4. Scroll to the bottom of the document just before the closing page tags, delete the text **[replace with function call]**, insert opening and closing script tags, and type **doTrans()** as the body of the script
This script, shown in Figure M-12, calls the doTrans() function when the page finishes loading in the browser window.

5. Check your document for errors and make any necessary changes, then save **Presentation transition element.htm** as a text document

Trouble?
If the logo does not appear gradually, your video card or monitor is probably not compatible with transitions.

6. Open **Presentation transition element.htm** in your Web browser
As Figure M-13 shows, the Nomad Ltd logo slowly fades into view as the page opens.

FIGURE M-11: Document code containing function

```
window.onload = grow;

function doTrans() {
        logo.filters.blendTrans.Apply();
        logo.style.visibility="visible";
        logo.filters.blendTrans.Play();
}
//-->
</SCRIPT>

</HEAD>
```

Function controlling transition effect

FIGURE M-12: Document code containing script to call function

```
<DIV>For more information on Nomad Ltd, please email our <A
HREF="MAILTO:relations@nomadltd.com">community relations
department</A>.</DIV>

</DIV>

<SCRIPT LANGUAGE="javascript">
<!--
doTrans()
//-->
</SCRIPT>

</BODY>
</HTML>
```
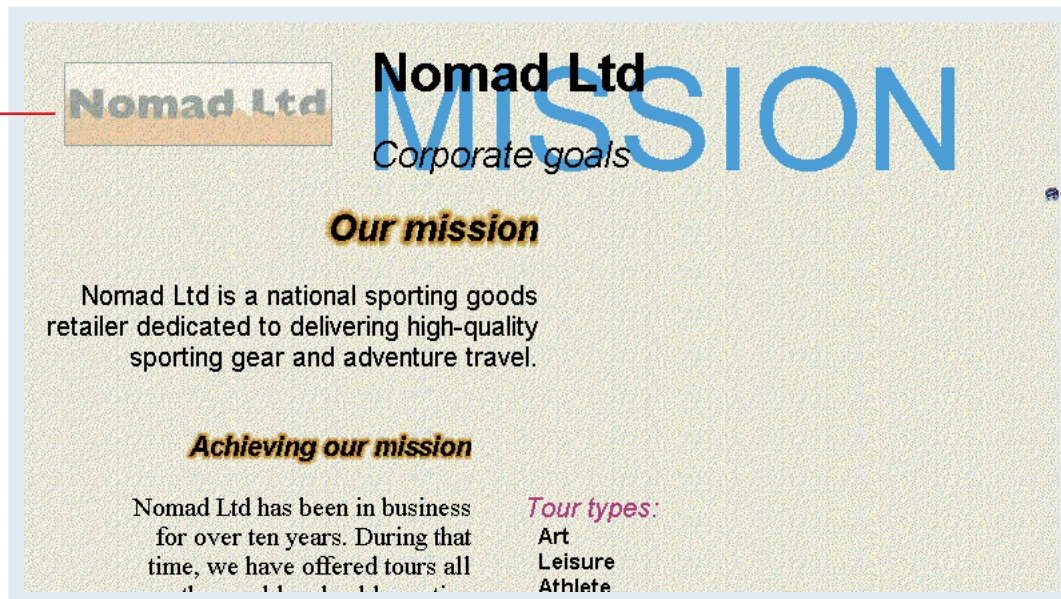
Script triggers doTrans() function after page loads

FIGURE M-13: Nomad Ltd logo showing blend transition

Logo gradually appearing

Earth graphic at start of animation

HTML

Creating a Slideshow

Presentation software, such as Microsoft PowerPoint, allows you to move through a related set of pages, or **slides**, by clicking a mouse button. It is easy to add this effect to Web pages with scripting. Although standard HTML hyperlinks can create a similar effect, DHTML features enable your users to advance by clicking anywhere on the page, rather than scrolling to locate and click a hyperlink. Also, by eliminating navigation-specific elements, you can keep your pages focused on the presentation topic and create a more unified design. ✎ Lydia has created the second page for her Web presentation. She wants to script the first page to open the second in response to a mouse click, which will allow Internet Explorer users to click anywhere on the Web page to advance to the second Web page.

Steps 1 2 3 4

1. Open the file **HTML M-6.htm** in your text editor, then save it as a text document with the filename **Presentation slideshow.htm**

2. Scroll down to the opening BODY tag, select and replace the text **[replace with event handler]** with **onClick="window.location.href='Presentation page 2.htm'"**
   This event handler changes the window's HREF, or page address, to "Presentation page 2.htm," the second page that Lydia prepared.

3. Scroll down below the text "Corporate goals," then select and replace the text **[replace with instruction text]** with the following script, pressing **[Enter]** at the end of each line:
   <DIV ID="instr">
   Click anywhere to advance to next slide.
   </DIV>
   This text tells users how to navigate through the presentation. Figure M-14 shows the Web page code containing the event handler and the instruction text. Lydia has already added an embedded style for the instructions and has edited the scaling code to reposition the text in smaller window sizes.

4. Check your document for errors, make any necessary changes, then save **Presentation slideshow.htm** as a text document

5. Open **Presentation slideshow.htm** in your Web browser, then click anywhere on the page
   The second presentation page opens, as shown in Figure M-15. By simply adding the event handler to all presentation pages except for the last one, Lydia can enable users to easily page through the presentation online.

FIGURE M-14: Web page code containing event handler and instruction text

```
</HEAD>

<BODY BACKGROUND="Egg shell.jpg" onClick="window.location.href='Presentation
page 2.htm'">

<DIV ID="logo">
<IMG SRC="nomad.jpg">
</DIV>

<DIV ID="bgword">
MISSION
</DIV>

<DIV ID="heading">
<H1>Nomad Ltd</H1>
<H2 CLASS="norm">Corporate goals</H2>
</DIV>

<DIV ID="instr">
Click anywhere to advance to next slide.
</DIV>

<DIV ID="earth">
<IMG SRC="earth.jpg" ID="earthpic" WIDTH=0>
</DIV>
```
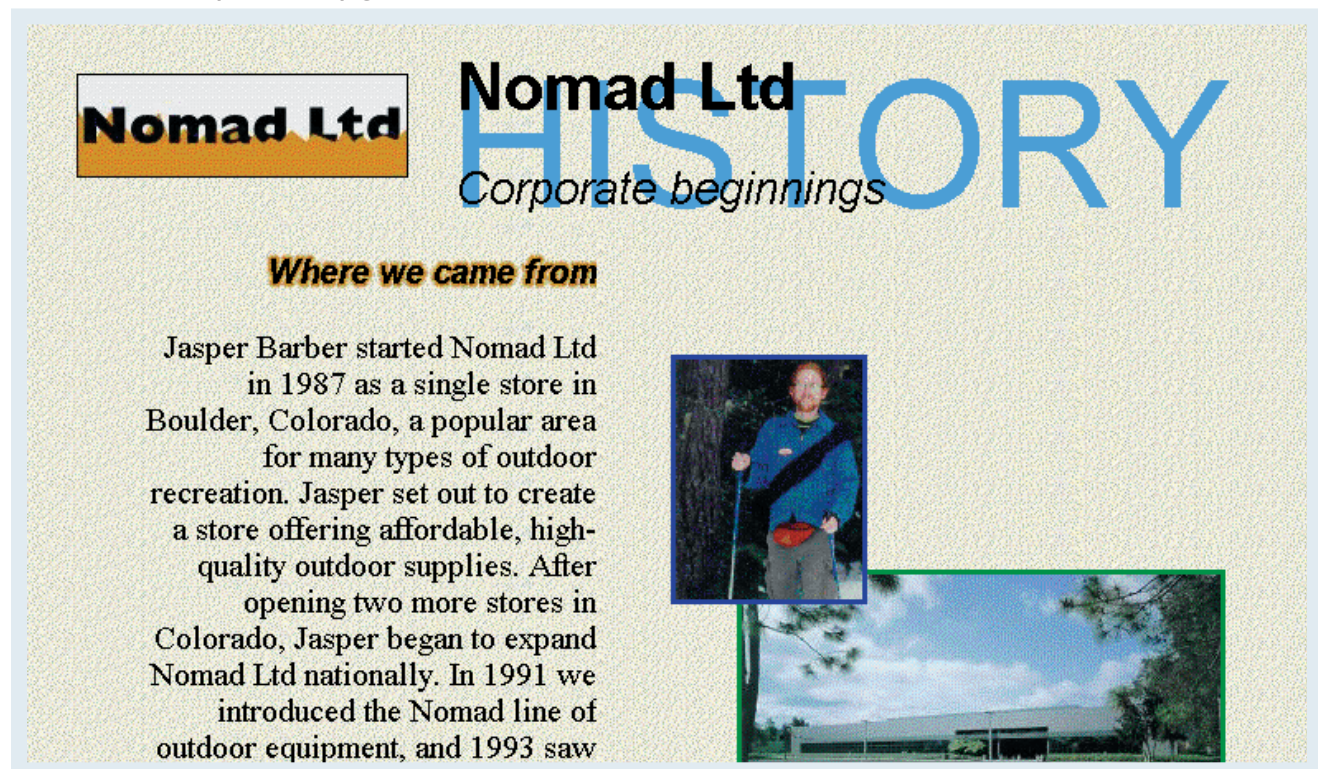
Event handler allows easy navigation to second page

Navigation instructions

FIGURE M-15: Second presentation page

Transitioning Between Pages

In addition to creating transition effects for specific elements on a Web page, you can apply transitions when opening or closing a page. In this situation, transitions can grab and hold a viewer's attention, and can help your page to stand out among pages a user has recently seen. Each Web page can trigger transitions upon opening and exiting, independent of the preceding or following page. As a final touch, Lydia decides she wants Internet Explorer 4 users to see a closing transition to each of the pages in her Web presentation. She starts by adding a closing transition that appears when the first page closes.

Steps 1 2 3 4

1. Open the file **HTML M-7.htm** in your text editor, then save it as a text document with the filename **Presentation page transition.htm**

2. Scroll down below the embedded style sheet in the page's head section and replace the text **[replace with META tag]** with **<META http-equiv="Page-Exit" CONTENT="RevealTrans(Duration=5,Transition=3)">**

   Figure M-16 shows the Web page code containing the META tag. Creating an interpage transition requires no scripting. Instead, you insert an HTML META tag in the page's head section, calling the transition and defining its properties. You can set the http-equiv property, which tells when the transition takes effect, to "Page-Enter" or "Page-Exit." You use the CONTENT property to specify the transition filter name and parameters, just as you do with the STYLE property for element transitions. Lydia uses the reveal transition's "Circle out" pattern, indicated by the Transition number 3. Table M-2 lists other reveal transitions and their number codes.

3. Check your document for errors, make any necessary changes, then save **Presentation page transition.htm** as a text document

4. Open **Presentation page transition.htm** in your Web browser, then click anywhere on the page

   The second presentation page opens in a circle spreading outward from the center of the browser window, as shown in Figure M-17. You can apply other transition patterns, as listed in Table M-2, to open or close pages.

5. Make and save changes in your text-editor program as needed, check changes in your Web browser program, close your Web browser program, then close your text editor program

Trouble?

Depending on your computer speed, you may see a small white circle in the center of the screen just before the transition starts.

FIGURE M-16: Web page code containing META tag

META tag inserted to control interpage transition

```
.sidebar {float: right; position: absolute; left: 375px; font-family: arial;
width: 200px}
</STYLE>

<META http-equiv="Page-Exit" CONTENT="RevealTrans(Duration=5,Transition=3)">

<SCRIPT LANGUAGE="javascript">
<!--
Nav4 = (document.layers) ? 1:0;
IE4 = (document.all) ? 1:0;

if(!IE4) {window.onerror=null}

function change() {
     if (document.body.clientWidth < 640) {
          bgword.style.fontSize="48pt"
          earth.style.width="25%"
     }
     else {
          bgword.style.fontSize="64pt"
          earth.style.width="35%"
     }
}

window.onresize=change
window.onload=change
```

FIGURE M-17: Web page closing with "Circle out" reveal transition

First presentation page

Second presentation page opening outward in circle

TABLE M-2: Reveal transition effects

reveal transition name	value	reveal transition name	value	reveal transition name	value
Box in	0	Vertical blinds	8	Split horizontal out	16
Box out	1	Horizontal blinds	9	Strips left down	17
Circle in	2	Checkerboard across	10	Strips left up	18
Circle out	3	Checkerboard down	11	Strips right down	19
Wipe up	4	Random dissolve	12	Strips right up	20
Wipe down	5	Split vertical in	13	Random bars horizontal	21
Wipe right	6	Split vertical out	14	Random bars vertical	22
Wipe left	7	Split horizontal in	15	Random	23

▶ Concepts Review

Label the advanced DHTML effects indicated on Lydia's Web page in Figure M-18.

Match each term with its description.

4. **Filter**
5. **Position animation**
6. **3D animation**
7. **Transition**
8. **Scaling**

a. Gradually changes element size, using the width and height properties
b. Gradually changes element appearance, becoming visible or hidden
c. Gradually changes element position, using top and left style properties
d. Fits page elements to lower screen resolution or smaller window size
e. Property that modifies element appearance in complex ways

Select the best answer from the list of choices.

9. **You can continually create new advanced DHTML features by combining basic features with**
   a. CSS.
   b. Dynamic style.
   c. Dynamic content.
   d. Scripts.

10. **Which of the following is *not* a filter?**
    a. Shadow
    b. Blur
    c. Animation
    d. Blend transition

11. **What is the advantage of using DHTML animation rather than other animation methods on the Web?**
    a. DHTML animation doesn't require scripts.
    b. DHTML animation uses few system resources.
    c. DHTML animation uses special software.
    d. DHTML animation uses no system resources.

12. **Which method will *not* change an element's appearance over a period of time?**
    a. Position animation
    b. 3D animation
    c. Glow filter
    d. Blend transition filter

13. **Which HTML tag do you use to implement an interpage transition?**
    a. <LINK>
    b. <META>
    c. <A>
    d. <SCRIPT>

14. **Which is an advantage of using animated GIFs rather than DHTML animation?**
    a. Animated GIFs use no system resources.
    b. Animated GIF display is not limited to fourth-generation browsers.
    c. Creating animated GIFs requires no special software.
    d. Animated GIFs can show 3D animation.

▶ Skills Review

1. Filter content.

    a. Open HTML M-8.htm in your Web browser and explore the page.

    b. Open HTML M-8.htm in your text editor and save a copy as Tours filter.htm.

    c. In the embedded style sheet, select the text [replace with histhead filter code], then press [Delete].

    d. Type height: 14pt; filter: glow(color=#6B8E23)

    e. Select the text [replace with second filter code] in the #tourshead style definition, then press [Delete].

    f. Type height: 12pt; filter: glow(color=#6B8E23, strength=3)

    g. Check your document for errors, make changes as necessary, then save Tours filter.htm.

    h. Open Tours filter.htm in your Web browser.

2. Scale content.

    a. Open HTML M-9.htm in your text editor, then save a copy as Tours scale.htm.

    b. Select the text [replace with scale script], then press [Delete].

    c. Enter the opening tags for a JavaScript script, then type the following script, pressing [Enter] at the end of each line:

```
function change() {
        if (document.body.clientWidth < 640) {
                bgword.style.fontSize="48pt"
        }
        else {
                bgword.style.fontSize="64pt"
        }
}

window.onresize=change
window.onload=change
```

    d. Enter the closing script tags, check your document for errors, make changes as necessary, then save Tours scale.htm.

    e. Open Tours scale.htm in your Web browser, then click the Restore Window button to decrease the size of the document window a fixed amount and note the changes to the background text in the heading. If necessary, drag the right edge of the window to the left to decrease the screen width until the change takes place.

    f. Click the browser maximize button.

3. Animate element position.

    a. Open the file HTML M-10.htm in your text editor, then save it as a text document with the filename Tours position animate.htm.

    b. Scroll down to the opening BODY tag, select the text [replace with event handler], then press [Delete].

    c. Type onLoad="slide()"

    d. Check your document for errors, make any necessary changes, then save as Tours position animate.htm.

    e. Open Tours position animate.htm in your Web browser.

4. Create 3D animation.

   a. Open the file HTML M-11.htm in your text editor, then save it as a text document with the filename Tours 3D animate.htm.

   b. Select the text [replace with 3D animation script] in the page's head section, press [Delete], then type the following script, pressing [Enter] at the end of each line:

```
function grow() {
        if (mtnpic.width<180) {
                x=window.setTimeout('grow()', 100)
                mtnpic.width=mtnpic.width + 10
        }
}
```

   c. In the BODY tag, delete the text [replace with event handler], and type onLoad="grow()"

   d. Scroll down to the IMG tag for the mountain graphic and replace the text [replace with width setting] with WIDTH=0

   e. Check your document for errors, make any necessary changes, then save as Tours 3D animate.htm.

   f. Open Tours 3D animate.htm in your Web browser.

5. Use transition elements.

   a. Open the file HTML M-12.htm in your text editor, then save a copy as Tours transition element.htm.

   b. In the #logo style specification in the page's embedded style sheet, replace the text [replace with style information] with visibility: hidden; filter: revealTrans(Transition=12, Duration=5)

   c. Scroll down to the end of the page's head section and replace the text [replace with transition function] with the following, pressing [Enter] at the end of each line:

```
function doTrans() {
        logo.filters.revealTrans.Apply();
        logo.style.visibility="visible";
        logo.filters.revealTrans.Play();
}
```

   d. Scroll to the bottom of the document just before the closing page tags, delete the text [replace with function call], insert opening and closing script tags, and type doTrans() as the body of the script.

   e. Check your document for errors, make any necessary changes, then save as Tours transition element.htm.

   f. Open Tours transition element.htm in your Web browser.

6. Create a slideshow.

   a. Open the file HTML M-13.htm in your text editor, then save it as a text document with the filename Tours slideshow.htm.

   b. Scroll down to the opening BODY tag and replace the text [replace with event handler] with onClick="window.location.href='Tours page 2.htm'"

   c. Scroll down below the text "Tours division", select and replace the text [replace with instruction text] with the following script, pressing [Enter] at the end of each line:

```
<DIV ID="instr">
Click anywhere to advance to next slide.
</DIV>
```

   d. Check your document for errors, make any necessary changes, then save Tours slideshow.htm.

   e. Open Tours slideshow.htm in your Web browser, then click anywhere on the page.

HTML

7. Transition between pages.

   a. Open the file HTML M-14.htm in your text editor, then save a copy as Tours page transition.htm.

   b. Scroll down below the embedded style sheet in the page's head section and replace the text [replace with META tag] with the following: <META http-equiv="Page-Exit" CONTENT="RevealTrans(Duration=5,Transition=10)">

   c. Check your document for errors, make any necessary changes, then save as Tours page transition.htm.

   d. Open Tours page transition.htm in your Web browser, then click anywhere on the page.

► Independent Challenges

1. The owners of the Green House plant store have seen transition filters in use on other Web pages and think this effect would make their pages more interesting. You decide to add the "random dissolve" reveal transition to the secondary heading on the "Popular supplies" page.

   To complete this independent challenge:

   a. Open the file HTML M-15.htm in your text editor, then save it as a text document with the filename Green House transition.htm.

   b. Add the following style to the embedded style sheet in the page's head section:

     #subhead {visibility: hidden; height: 16pt; filter: revealTrans(Transition=12, Duration=5)}.

   c. Scroll down to the end of the page's head section and replace the text [replace with transition function] with the following script, pressing [Enter] at the end of each line. Be sure to include opening and closing script tags.

```
function doTrans() {
        subhead.filters.revealTrans.Apply();
        subhead.style.visibility="visible";
        subhead.filters.revealTrans.Play();
}
```

   d. In the opening BODY tag, replace the text [replace with function call] with onLoad="doTrans()".

   e. Check your document for errors, make any necessary changes, then save Green House transition.htm.

   f. Open Green House transition.htm in your Web browser and verify that the subhead text appears slowly, in a random dissolve pattern.

2. You've long wanted to add an animated bus graphic to the home page you created for Sandhills Regional Public Transit (SRPT). Now that you know how to animate the position of Web page objects, you want to script this feature at the top of each SRPT Web page.

   To complete this independent challenge:

   a. Open the file HTML M-16.htm in your text editor, then save it as a text document with the filename SRPT animated.htm.

   b. To create space at the top of the page and to avoid overlapping elements, change the "top" values in the embedded style sheet for the following styles to the values indicated:

     #bus: 75px
     #head1: 60px
     #head2: 300px
     #instr: 75px

   c. Add the following entry to the embedded style sheet for the moving graphic:

     #move {position: absolute; left: -1000px}

   d. You have already inserted the slide() function for this task. Look at the script in the page's HEAD section, noting the differences from the one you used in this unit. Try to predict how this script will behave differently, if at all.

e. Add the following event handler to the opening BODY tag:
   onLoad="slide()"

f. Add the following IMG tag for the moving bus graphic immediately after the page's opening BODY tag:
   <IMG SRC="busmove.jpg" ID="move">

g. Check your document for errors, then save SRPT animated.htm.

h. Open SRPT animated.htm in your Web browser and verify that the new bus graphic moves across the page without overlapping other elements.

3. The public-relations coordinator of Community Public School Volunteers wants to make sure the organization's Web page has an appealing layout, regardless of the computer's screen resolution or window size. Rather than using a scaling script, however, you decide to specify element dimensions in percentages to ensure uniform appearance.
   To complete this independent challenge:

a. Open the file HTML M-17.htm in your text editor, then save it as a text document with the filename CPSV scale.htm.

b. In the #logo definition in the page's embedded style sheet, set the width value to 90%.

c. In the #box definition, change the width value to 75% and the left value to 12.5% (these settings ensure that the box is centered at any window size).

d. In the #instr definition, change the left value to 5% and the width value to 90% (these settings ensure that the text is centered in the box).

e. Check your changes, then save as CPSV scale.htm.

f. Preview CPSV scale.htm in your Web browser, then use your text editor to make any necessary changes, as well as any positioning and sizing changes that you think would improve the page's layout.

WEB WORK

4. In addition to the filters and transitions that you've used in this unit, Internet Explorer 4 and 5 offer large selections of each of these features. To complete this independent challenge, connect to the Internet and locate a Web page that explains specific filters and transitions in detail. You can find this information on the Microsoft Web site, as well as on other DHTML sites. Find and print information on one standard filter and one Reveal transition filter that you have not used before. Make sure the information you print includes the name of the filter, the syntax for using it in your style specifications, and an explanation of any special parameters that it allows you to set. Then create a simple Web page named Unit M IC 4.htm, which contains one element. Apply the standard filter to the element and apply the Reveal transition filter to the page exit. Submit the printouts from your research and your file containing the filter and transition to your instructor.

▶ Visual Workshop

The owner of Touchstone Booksellers would like users to see an interesting effect when they first view his page. You decide to try an interpage transition that takes effect when the page opens. Save a copy of the file HTML M-18.htm as Touchstone transition.htm. Insert the necessary code to add the "Random bars vertical" transition, shown in Figure M-19. Use Table M-2 to look up the Transition value for this effect. The http-equiv value for this transition should be "Page-Enter" for the transition to occur as the page first loads. To test your changes, save your file, then open the file "Touchstone open.htm" in your Web browser, and click the link "Touchstone home page" to open the page you created.

FIGURE M-19